

(12) **United States Patent**  
**Ulasen et al.**

(10) **Patent No.:** **US 12,165,352 B2**  
(45) **Date of Patent:** **Dec. 10, 2024**

(54) **SYSTEMS AND METHODS FOR DETERMINING ENVIRONMENT DIMENSIONS BASED ON ENVIRONMENT POSE**

(58) **Field of Classification Search**  
CPC ..... G06T 7/70; G06T 2207/20081; G06T 2207/20084; G06T 2207/30228;  
(Continued)

(71) Applicant: **Acronis International GmbH**,  
Schaffhausen (CH)  
(72) Inventors: **Sergey Ulasen**, Singapore (SG);  
**Alexander Snorkin**, Dubai (AE);  
**Andrey Adaschik**, Istanbul (TR);  
**Artem Shapiro**, Dnipro (UA); **Vasyl Shandyba**, Dnipro (UA); **Serg Bell**,  
Costa Del Sol (SG); **Stanislav Protasov**, Singapore (SG)

(56) **References Cited**  
U.S. PATENT DOCUMENTS  
2020/0364509 A1\* 11/2020 Weinzaepfel ..... G06F 18/214  
FOREIGN PATENT DOCUMENTS  
CN 108875524 A \* 11/2018 ..... G06K 9/00597  
CN 111652155 A \* 9/2020  
(Continued)

(73) Assignee: **Acronis International GmbH**,  
Schaffhausen (CH)  
(\*) Notice: Subject to any disclaimer, the term of this  
patent is extended or adjusted under 35  
U.S.C. 154(b) by 351 days.

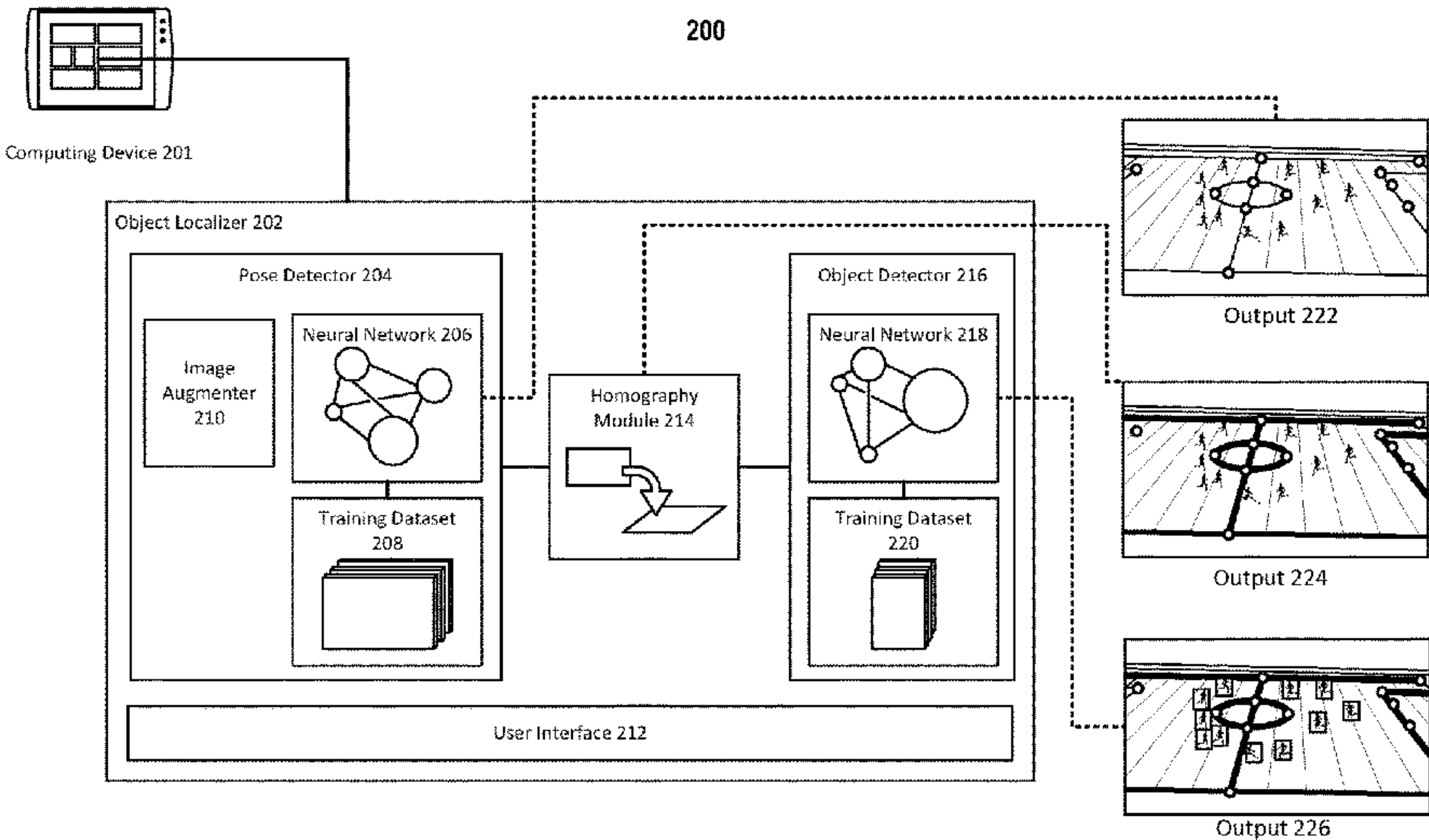
*Primary Examiner* — Xin Jia  
(74) *Attorney, Agent, or Firm* — ArentFox Schiff LLP;  
Michael Fainberg

(21) Appl. No.: **17/734,399**  
(22) Filed: **May 2, 2022**  
(65) **Prior Publication Data**  
US 2022/0405954 A1 Dec. 22, 2022

**Related U.S. Application Data**  
(60) Provisional application No. 63/210,599, filed on Jun.  
15, 2021.  
(51) **Int. Cl.**  
**G06K 9/00** (2022.01)  
**G06N 3/08** (2023.01)  
**G06T 7/70** (2017.01)  
(52) **U.S. Cl.**  
CPC ..... **G06T 7/70** (2017.01); **G06N 3/08**  
(2013.01); **G06T 2207/20081** (2013.01); **G06T**  
**2207/20084** (2013.01); **G06T 2207/30228**  
(2013.01)

(57) **ABSTRACT**  
Disclosed herein are systems and method for determining environment dimensions based on environment pose. In one aspect, the method may include training, with a dataset including a plurality of images featuring an environment and labelled landmarks in the environment, a neural network to identify a pose of an environment. The method may comprise receiving an input image depicting the environment, generating an input tensor based on the input image, and inputting the input tensor into the neural network, which may be configured to generate an output tensor including a position of each identified landmark, a confidence level associated with each position, and a pose confidence score. The method may include calculating a homography matrix between each position in the output tensor along a camera plane and a corresponding position in an environment plane in order to output an image that visually connects each landmark along the environment plane.

**17 Claims, 5 Drawing Sheets**



(58) **Field of Classification Search**

CPC ..... G06T 7/60; G06T 7/73; G06N 3/08;  
G06N 3/0464; G06N 3/09

See application file for complete search history.

(56) **References Cited**

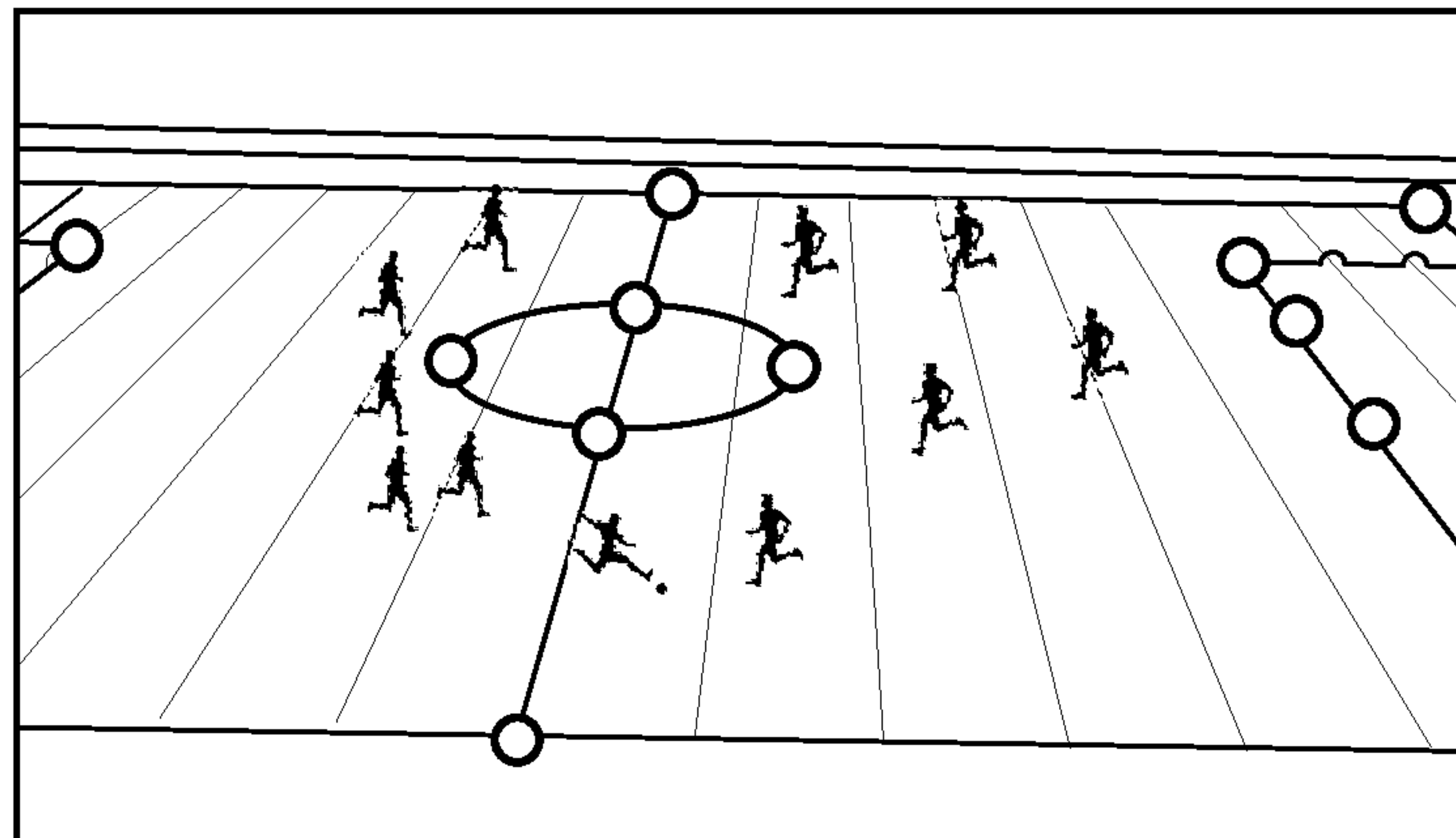
FOREIGN PATENT DOCUMENTS

CN	111652929	A	*	9/2020
CN	111723687	A	*	9/2020

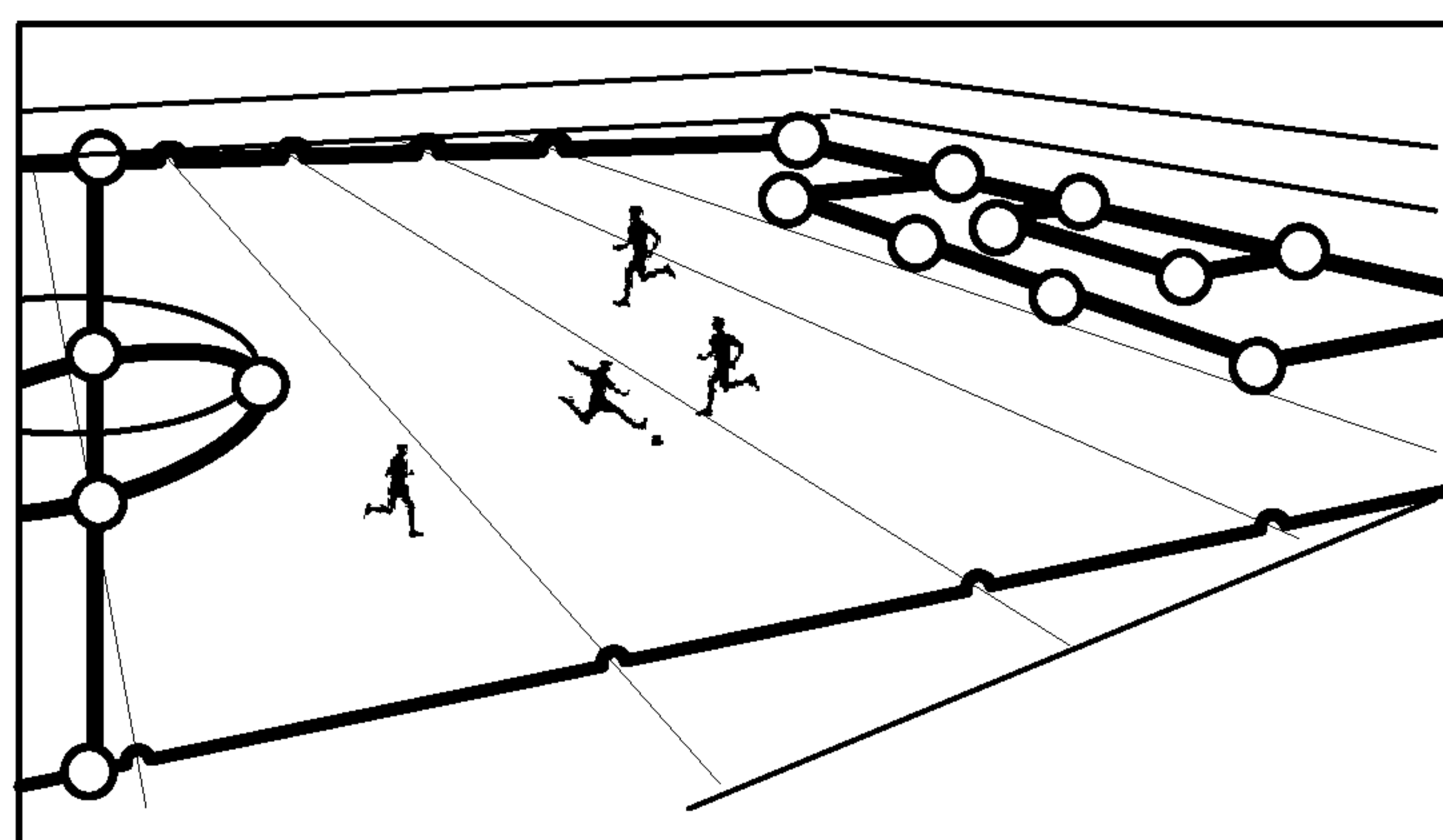
\* cited by examiner

100

Training Image



Key Landmarks Detection



Pose Detection Architecture

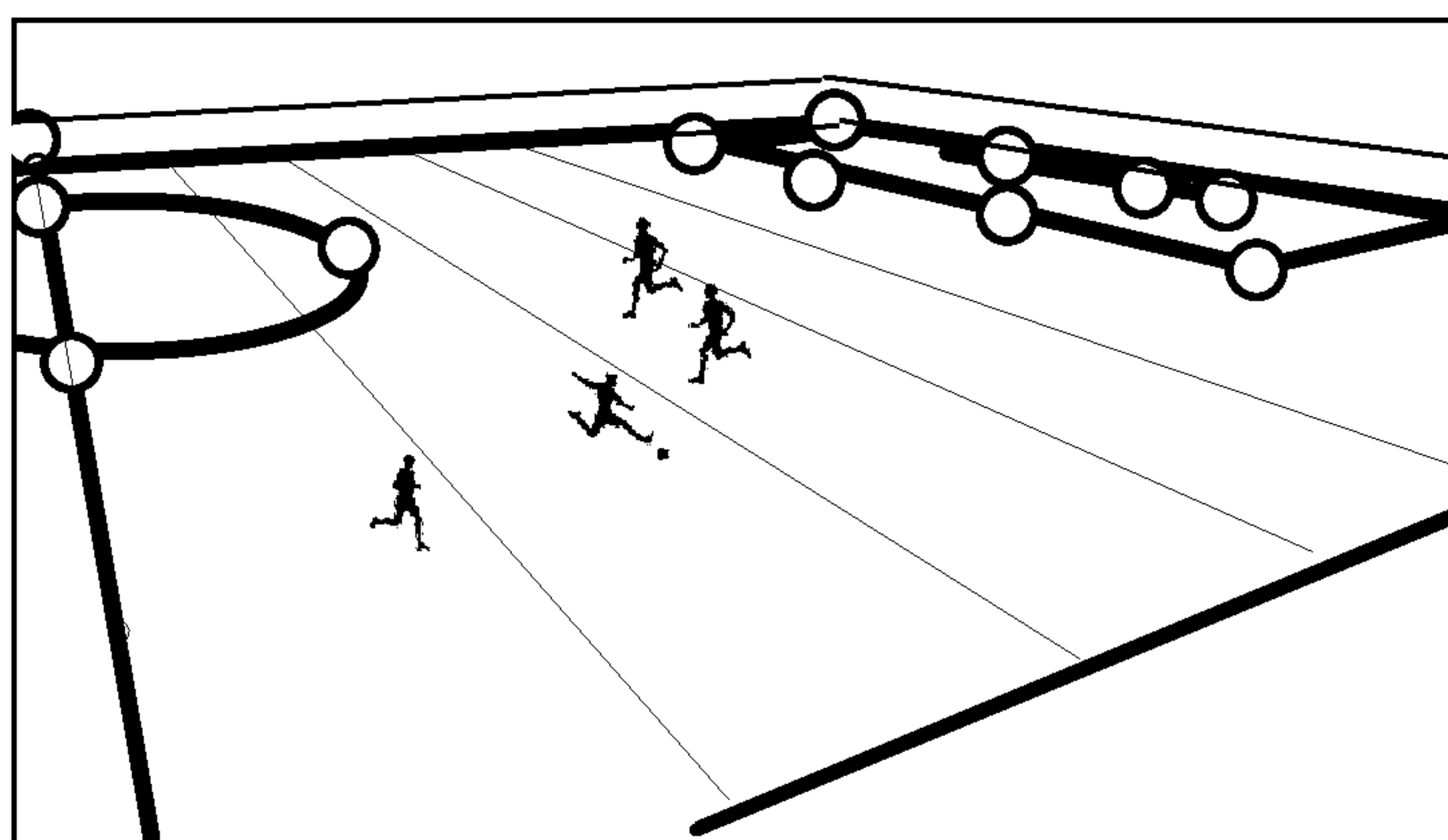


Fig. 1

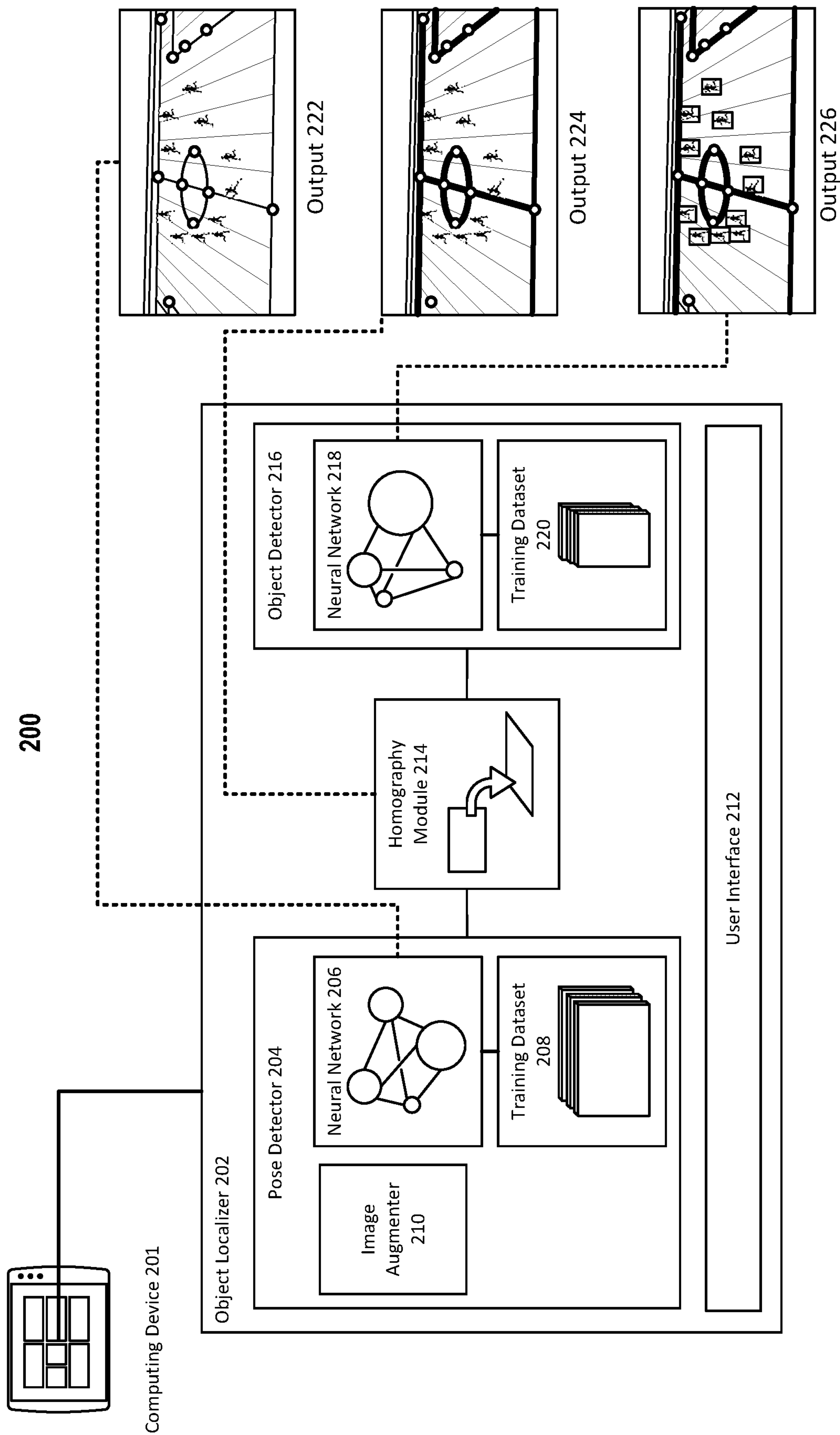


Fig. 2

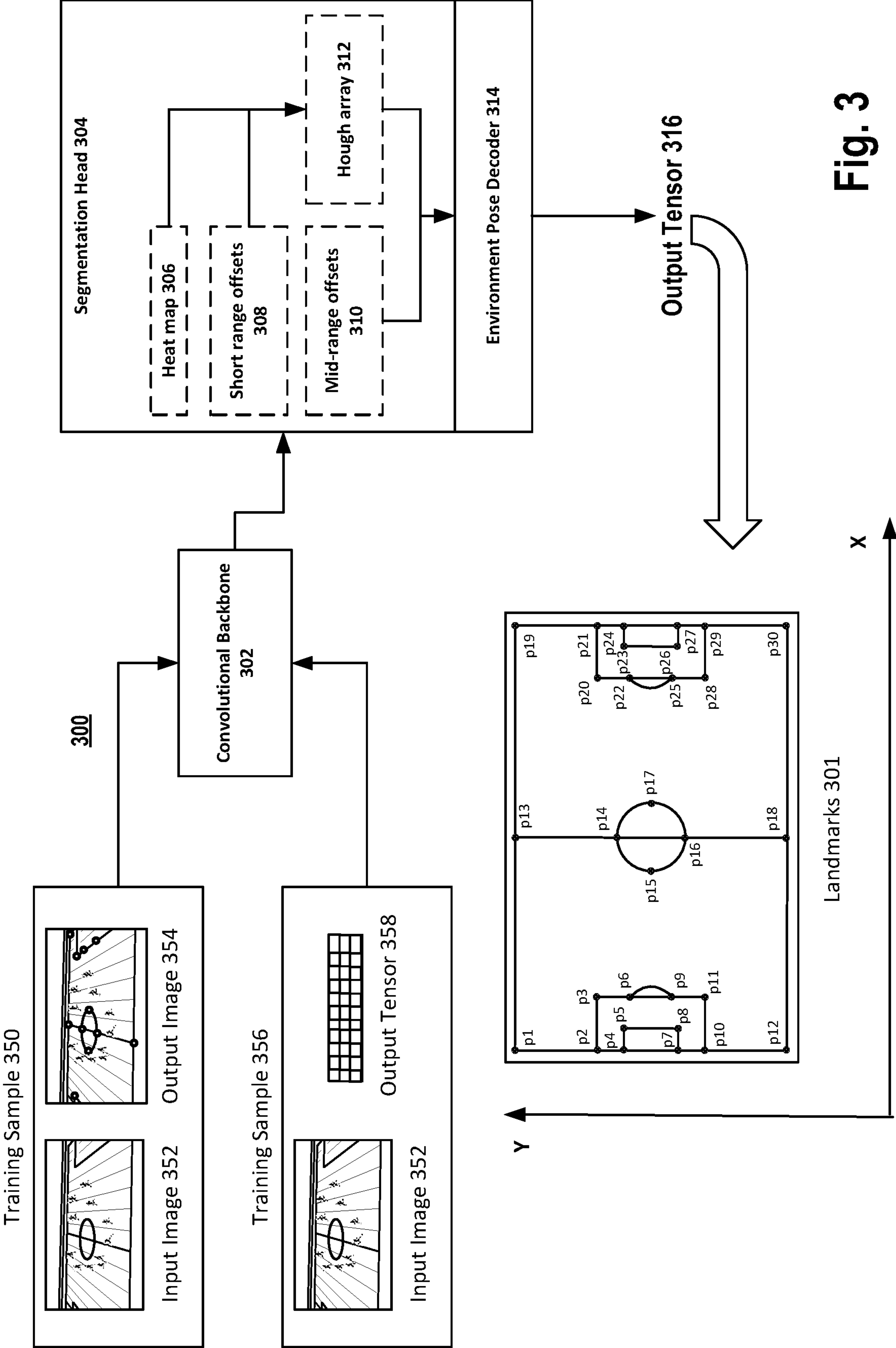
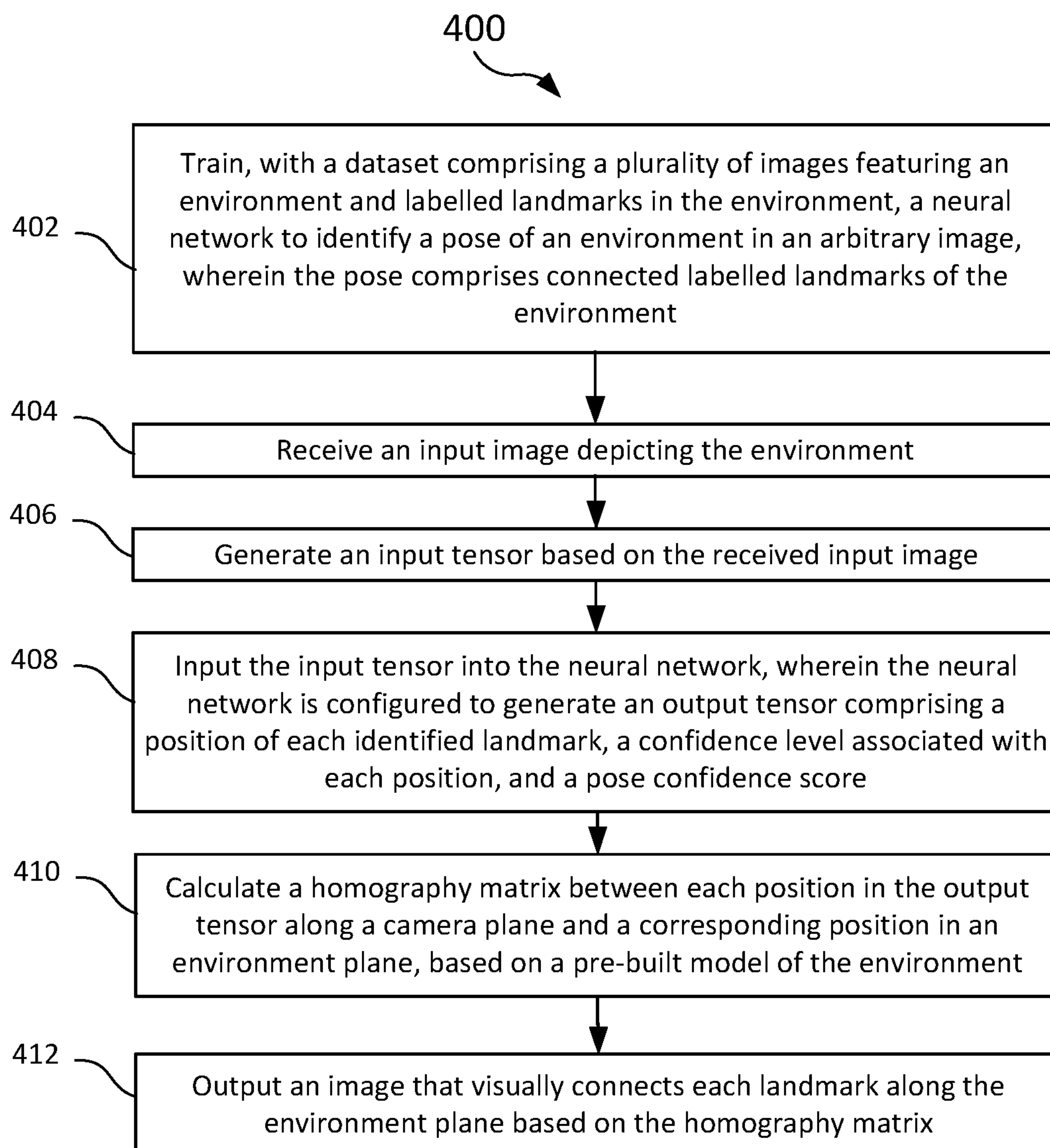


Fig. 3



**Fig. 4**

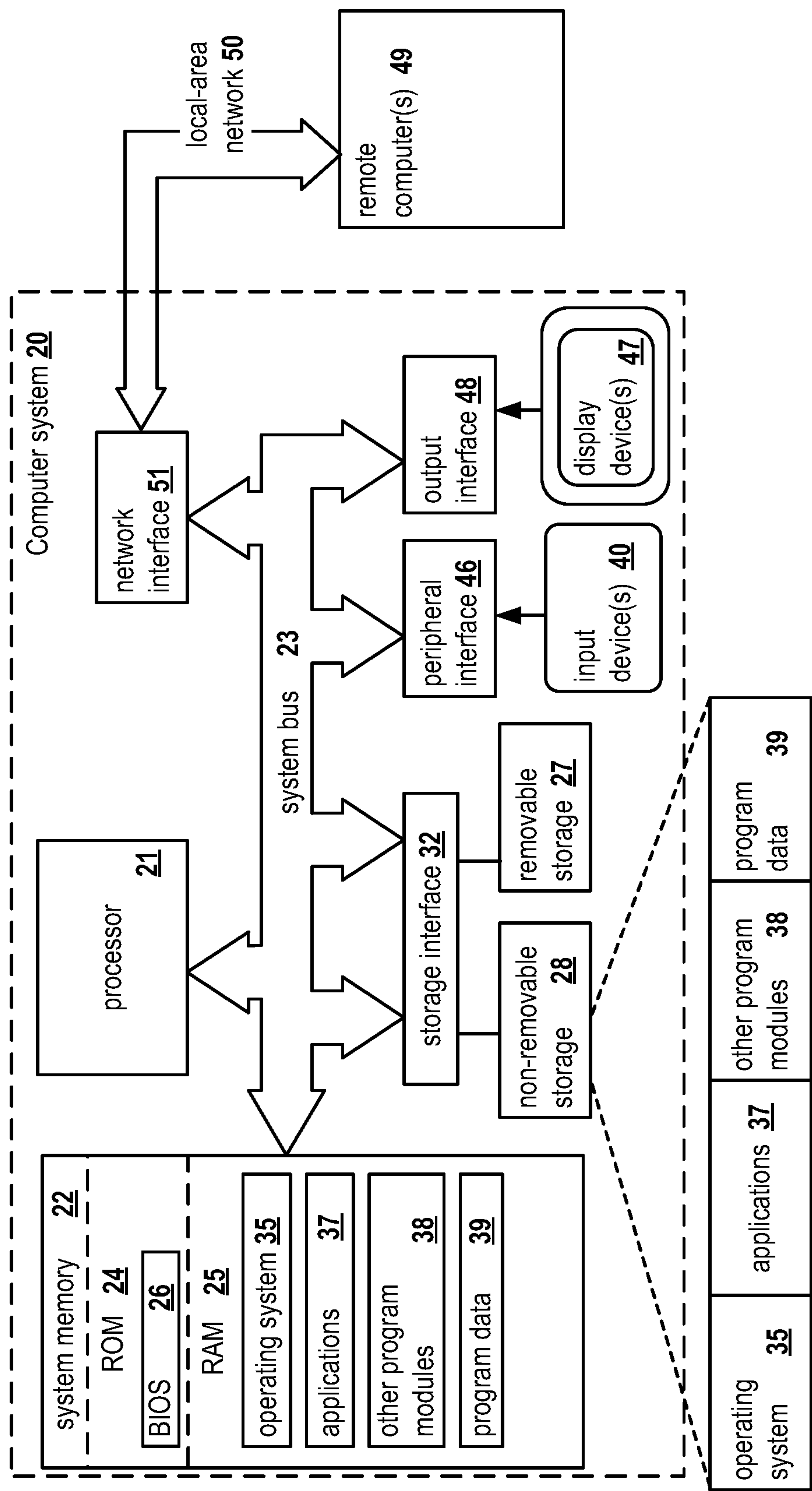


Fig. 5



## 1

# SYSTEMS AND METHODS FOR DETERMINING ENVIRONMENT DIMENSIONS BASED ON ENVIRONMENT POSE

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 63/210,599, filed Jun. 15, 2021, which is herein incorporated by reference.

## FIELD OF TECHNOLOGY

The present disclosure relates to the field of computer vision, and, more specifically, to systems and methods for determining environment dimensions based on environment pose.

## BACKGROUND

When localizing objects using computer vision and machine learning, object detection is usually described relative to the dimensions of the two-dimensional input image. For example, the location of a detected object may be described as pixel coordinates. This information is useful, but not relevant if a user wants to know the location of the object relative to the environment that the object is in. Consider an example where an input image is of a broadcast view of a soccer match. A user may be interested in identifying the players on the field and determining their location on the field (rather than on the image). In order to determine the location of the objects on the field, such as the players, the dimensions of the field need to be determined.

One approach to determining the dimensions is key point detection, which involves identifying points in an image that will always have the same real-life positions relative to one another (e.g., the center point of an official soccer pitch will be the same distance from the soccer net irrespective of which official soccer pitch it is). Although key point detection enables the dimensions of a location to be estimated, depending on the camera perspective, the dimensions may be inaccurate.

Accordingly, there is a need for improved dimension determination.

## SUMMARY

To address these shortcomings, aspects of the disclosure describe methods and systems for determining environment dimensions based on environment pose.

In one exemplary aspect, the techniques described herein relate to a method for determining environment dimensions based on environment pose, the method including: training, with a dataset including a plurality of images featuring an environment and labelled landmarks in the environment, a neural network to identify a pose of an environment in an arbitrary image, wherein the pose includes connected labelled landmarks of the environment; receiving an input image depicting the environment; generating an input tensor based on the received input image; inputting the input tensor into the neural network, wherein the neural network is configured to generate an output tensor including a position of each identified landmark, a confidence level associated with each position, and a pose confidence score; calculating a homography matrix between each position in the output tensor along a camera plane and a corresponding position in

## 2

an environment plane, based on a pre-built model of the environment; and outputting an image that visually connects each landmark along the environment plane based on the homography matrix.

In some aspects, the techniques described herein relate to a method, wherein a camera perspective of the input image does not match any of the camera perspectives of the plurality of images in the dataset.

In some aspects, the techniques described herein relate to a method, wherein the neural network includes: a convolutional backbone with feature extraction layers, and a segmentation head.

In some aspects, the techniques described herein relate to a method, wherein the pre-built model of the environment is indicative of distances between each landmark in the environment.

In some aspects, the techniques described herein relate to a method, wherein the neural network is further configured to determine a heat map for each position of each identified landmark, wherein the heat map represents an area in which the identified landmark may be in the input image.

In some aspects, the techniques described herein relate to a method, wherein the neural network optimizes the loss using stochastic gradient descent.

In some aspects, the techniques described herein relate to a method, wherein the input image is a video frame of a livestream, and wherein the neural network determines environment dimensions in real-time.

In some aspects, the techniques described herein relate to a method, wherein the environment is a sports field and the labelled landmarks are locations on the sports field.

It should be noted that the methods described above may be implemented in a system comprising a hardware processor. Alternatively, the methods may be implemented using computer executable instructions of a non-transitory computer readable medium.

In some aspects, the techniques described herein relate to a system for determining environment dimensions based on environment pose, the system including: a hardware processor configured to: train, with a dataset including a plurality of images featuring an environment and labelled landmarks in the environment, a neural network to identify a pose of an environment in an arbitrary image, wherein the pose includes connected labelled landmarks of the environment; receive an input image depicting the environment; generate an input tensor based on the received input image; input the input tensor into the neural network, wherein the neural network is configured to generate an output tensor including a position of each identified landmark, a confidence level associated with each position, and a pose confidence score; calculate a homography matrix between each position in the output tensor along a camera plane and a corresponding position in an environment plane, based on a pre-built model of the environment; and output an image that visually connects each landmark along the environment plane based on the homography matrix.

In some aspects, the techniques described herein relate to a non-transitory computer readable medium storing thereon computer executable instructions for determining environment dimensions based on environment pose, including instructions for: training, with a dataset including a plurality of images featuring an environment and labelled landmarks in the environment, a neural network to identify a pose of an environment in an arbitrary image, wherein the pose includes connected labelled landmarks of the environment; receiving an input image depicting the environment; generating an input tensor based on the received input image;



inputting the input tensor into the neural network, wherein the neural network is configured to generate an output tensor including a position of each identified landmark, a confidence level associated with each position, and a pose confidence score; calculating a homography matrix between each position in the output tensor along a camera plane and a corresponding position in an environment plane, based on a pre-built model of the environment; and outputting an image that visually connects each landmark along the environment plane based on the homography matrix.

The above simplified summary of example aspects serves to provide a basic understanding of the present disclosure. This summary is not an extensive overview of all contemplated aspects, and is intended to neither identify key or critical elements of all aspects nor delineate the scope of any or all aspects of the present disclosure. Its sole purpose is to present one or more aspects in a simplified form as a prelude to the more detailed description of the disclosure that follows. To the accomplishment of the foregoing, the one or more aspects of the present disclosure include the features described and exemplarily pointed out in the claims.

### BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more example aspects of the present disclosure and, together with the detailed description, serve to explain their principles and implementations.

FIG. 1 is a diagram illustrating outputs from a key point detection approach and a pose detection approach.

FIG. 2 is a block diagram illustrating a system for determining environment dimensions based on environment pose.

FIG. 3 is a block diagram illustrating an exemplary neural network structure.

FIG. 4 illustrates a flow diagram of a method for determining environment dimensions based on environment pose.

FIG. 5 presents an example of a general-purpose computer system on which aspects of the present disclosure can be implemented.

### DETAILED DESCRIPTION

Exemplary aspects are described herein in the context of a system, method, and computer program product for determining environment dimensions based on landmark detection. Those of ordinary skill in the art will realize that the following description is illustrative only and is not intended to be in any way limiting. Other aspects will readily suggest themselves to those skilled in the art having the benefit of this disclosure. Reference will now be made in detail to implementations of the example aspects as illustrated in the accompanying drawings. The same reference indicators will be used to the extent possible throughout the drawings and the following description to refer to the same or like items.

FIG. 1 is a diagram illustrating outputs from a key point detection approach and a pose detection approach. As discussed in the background, simply using key point detection will not enable proper dimension determination of a location. Consider a training dataset that comprises a plurality of training images depicting soccer pitches. As depicted in the first image of FIG. 1, a camera perspective of the training images may be the typical broadcast view of a game. When key point detection (or key landmarks detection) is used, the

camera perspective can cause detection inaccuracies. For example, the second image in FIG. 1 is a view captured when the camera is slightly tilted. Accordingly, the center field key points are also tilted. This is because the key point detection approach emphasizes spacing between key points to approximate positions (i.e., if key point X is in position 1, key point Y will be 4 meters directly to the left at position 2 because that is how the training images are set up). When the spacing becomes distorted due to camera perspective, tilt, and lens differences, the rigid spacing will cause incorrect positioning when using key point detection.

The present disclosure discusses a pose detection architecture whose output is resilient to such distortions. As shown in the third image of FIG. 1, despite the camera tilt (which causes a distortion of the soccer pitch), the pitch is properly mapped (i.e., the dimensions are correct). Due to the difference in architecture between pose and key point approaches, the present disclosure's systems and methods allow for a proper determination of environment/location dimensions even when a camera perspective of the input image does not match any of the camera perspectives of the training images used to train the neural network of the exemplary aspect.

FIG. 2 is a block diagram illustrating system 200 for determining environment dimensions based on environment pose. In an exemplary aspect, system 200 includes computing device 201 (e.g., a server, a smartphone, etc.) that executes object localizer 202. Object localizer 202 may be a software that is configured to receive an input image and output a version of the input image that highlights the objects in the input image, identifies an environment in the input image, and/or provides locations of the objects relative to the environment.

Object localizer 202 comprises various modules including pose detector 204, homography module 214, and object detector 216. User interface 212 accepts an input image and provides, via object localizer 202, an output image (e.g., one of outputs 222, 224, and 226 depending on the output the user wants). In some aspects, neural networks 206 and 218 and training datasets 208 and 220 may be stored on a different device than computing device 201. Computing device 201 may be a computer system (described in FIG. 5) such as a smartphone. If the neural networks 206 and 218 and/or training datasets 208 and 220 are stored on a different device (e.g., a server), computing device 201 may communicate with the different device to acquire information about the structure of the neural networks, code of neural networks, images in the training datasets, etc. This communication may take place over a network (e.g., the Internet). For example, object localizer 202 may be split into a thin client application and a thick client application. A user may provide an input image via user interface 212 on computing device 201. Interface 212, in this case, is part of the thin client. Subsequently, the input image may be sent to the different device comprising the thick client with the neural networks and the training datasets. Neural network 218 may yield output image 226 and transmit it to computing device 201 for output via user interface 212.

Pose detector 204 comprises image augments 210, which is a module that receives an input image (e.g., depicting a soccer match broadcast frame) from user interface 212 and augments the input image for processing via neural network 206, which detects landmarks in the input image. For



## 5

example, image augments 210 may convert the input image into a grayscale image, may crop the input image, rotate the input image, resize the input image, etc. Image augments 210 may also generate a tensor representation of the input image after any combination of augmentations.

FIG. 3 is a block diagram illustrating an exemplary neural network structure 300 (e.g., of neural network 206 used for pose detection). Structure 300 comprises a convolutional neural network backbone 302. Backbone 302 includes the feature extraction layers that receive an input image and generate feature maps containing high-level summarized information. Example feature extraction layers may be convolutional and pooling layers. For example, backbone 302 may follow the structure of known image classifying neural networks such as Resnet, VGG-16, EfficientNet, etc.).

When passing the input image through neural network 206, output 222 may be generated. Neural network 206 may be trained using training dataset 208. Training dataset 208 may comprise a plurality of images depicting an environment and certain landmarks in the environment.

Consider an example in which the input image is a frame of a real-time video stream depicting multiple objects. This video stream may be of a soccer match and the multiple objects may include a soccer ball and humans (e.g., players, coaches, staff, fans, etc.). As shown in FIG. 1, the images may be a far-view of the soccer field (e.g., a broadcast view). Training dataset 220 may include a plurality of images, each depicting multiple objects. Training dataset 208 may include images of the environment (i.e., the soccer field) with labelled key points signifying landmarks.

For example, landmarks 301 in FIG. 3 contains 30 points (p1-p30) that, when connected, provide structure to the soccer field (e.g., the four points p14, p15, p16, p17 in the middle form a circle, the multiple points p12, p18, p30 along the edge of the field, etc.). These points are manually identified as landmarks in training images. A training sample for neural network 206 may take two formats. For example, the training sample may be training sample 350, which includes input image 352 and output image 354. Output image 354 includes annotated points in specific locations of input image 352. These points represent landmarks corresponding to landmarks 301. In another format, training sample 356 may include output tensor 358 (used interchangeably with output vector) of shape [N, 2]. N represents the number of landmark points that can potentially be identified. Following the example of a soccer field, where N equals 30, output tensor 358 has the shape [30, 2]. This tensor has 30 rows (1 per key point) and 3 columns: x, y. Here, x and y represent coordinates of a point on frame. In output image 354, only 11 of the 30 landmarks are identified. The remaining 19 are outside of the visible portion of the soccer field. A corresponding output tensor 358 may be organized as:

TABLE 1

Data inside a Training Output Vector/Tensor															
Landmark Points															
	P1	P2	P3	P4	P5	...	P14	P15	P16	P17	...	P27	P28	P29	P30
X	0	0	100	0	0	...	500	400	500	600	...	0	0	0	0
Y	0	0	800	0	0	...	600	500	400	500	...	0	0	0	0

## 6

Here, the x and y coordinates represent pixel values. For example, point p14 has coordinates (500, 600) signifying a pixel that is 500 pixels from the left most pixel of input image 352 and 600 pixels from the bottom most pixel of input image 352. For points that have coordinates (0,0), the point is off of the image. For example, point p1 has coordinates (0, 0) and referring to landmarks 301 and output image 354, it can be seen that point p1 is not visible in input image 352.

There are three types of key points for each image: (1) key points outside the frame for which coordinates are not labeled, (2) key points inside the frame for which coordinates are labeled, and (3) key points outside frame for which coordinates are labeled. The third type appears during augmentation.

Neural network 206 may receive a plurality of training samples such as training sample 356 and learn features associated with each point of landmarks 301. Because each input image 352 includes a different view of the environment, some points may be visible while others may not be visible. In some images, all points may be visible. Based on the relative positions of each point, neural network 206 may be trained to estimate an output tensor 316 for any given input image.

On a more technical level, in neural network 206, convolutional backbone 302 extracts features from input image 352. These features are sent to segmentation head 304, which outputs output tensor 316. Output tensor 316 produced by segmentation head 304 is compared with output tensor 358 (i.e., the real values) and the neural network is trained based on the comparison (i.e., how far off the guess of segmentation head 304 is).

Unlike output tensor 358, which includes the true position values, output tensor 316 may be shape [N, 4]. N still represents the number of landmark points that can potentially be identified. Following the example of a soccer field, where N equals 30, output tensor 316 has the shape [30, 4]. This tensor has 30 rows (1 per key point) and 4 columns: x, y, position confidence score, and pose confidence score. Here, x and y represent predicted coordinates of a point on frame, and the position confidence score is a probability that the likelihood that the (x,y) predicted coordinates are equal to the real coordinates, and the pose confidence score is a mean value of the position confidence scores. A corresponding output tensor 316 may be organized as:



TABLE 2

Data inside Output Vector/Tensor of Arbitrary Image															
	Landmark Points														
	P1	P2	P3	P4	P5	...	P14	P15	P16	P17	...	P27	P28	P29	P30
X	0	0	99	0	0	...	500	405	450	500	...	0	0	0	0
Y	0	0	780	0	0	...	610	550	470	500	...	0	0	0	0
Position	90	90	90	88	71	...	78	89	74	72	...	91	95	98	90
Confidence Score (%)															
Pose	84	84	84	84	84	...	84	84	84	84	...	84	84	84	84
Confidence Score (%)															

15

More specifically, segmentation head **304** of neural network **206** is configured to determine, from the extracted features, a heat map **306** for each key point (used interchangeably with landmark point), short range offsets **308**, mid-range offsets **310**, and Hough array **312**. Hough array **312** is generated using Hough voting on heat map **306** and short range offsets **308**. Segmentation head **304** further comprises an environment pose decoder **314**, which takes mid-range offsets **310** and Hough array **312**, and outputs coordinates of the key points.

Suppose that disk  $D(p)$  is a disk of radius  $R$  centered around point  $p$  in an image. In some aspects,  $p_k$  is key point where a landmark may reside. For every landmark point, segmentation head **304** predicts a heat map  $H(p_k)$  such that  $H(p_k)=1$  if  $p \in D(p_k)$ , otherwise  $H(p_k)=0$ . Thus, for each respective key point, segmentation head **304** predicts a disk of radius  $R$  around the respective key point of the environment.

In some aspects, the radius is set to a predetermined number of pixels (e.g., 10). During training, segmentation head **304** calculates the heatmap loss as the average L1 loss for image positions and back-propagates the heatmap loss across the image of the environment. In some aspects, this loss is minimized using an optimizer such as stochastic gradient descent without the long range offsets found in conventional pose detection neural networks.

Segmentation head **304** predicts short-range offsets **308** which comprise vectors  $S(d)$ . At each point  $d$  within the key point disks, the short-range 2-D offset vector  $S(d)$  points from the point  $d$  to the  $i$ -th keypoint of the environment. Thus, segmentation head **304** generates  $N$  vector fields (where  $N$  is the number of landmark points) and solves a regression problem at each point  $d$ . During training, segmentation head **304** penalizes the short-range offset prediction errors with the L1 loss.

Segmentation head **304** then aggregates the short-range offsets and the heatmap using hough voting to generate Hough array **312**. More specifically, each point  $p$  casts a vote to each key point  $i$ . In some aspects, each vote has a weight equal to an activation probability of the point  $p$ . The local maxima in the Hough arrays serve as candidate positions for key points.

Segmentation head **304** then generates mid-range offsets **310** to group together the key points. Environment pose decoder **314** connects the key points using the mid-range offsets **310** and Hough array **312**. Segmentation head **304** ultimately estimates the position of each landmark (i.e., keypoint) and provides a confidence level of the estimation. The mean value of the confidence levels is the pose confidence score.

The output of neural network **206** may be a visual output (e.g., output **222**) with highlighted landmarks or a numerical output (e.g., a tensor of size  $[N, 4]$ ). When the output is numerical, the tensor may be input into homography module **214**. Homography module **214** creates the structure of the environment for which landmarks were identified. Homography module **214** may have environment models that indicate the distance between landmarks. For example, homography module **214** may store a soccer field model shown in FIG. 2, which lists the distances between the key points. Homography module **214** may then calculate a homography matrix between key points on a camera plane and the same key points on a field plane. Using the calculated homography and the environment models, any point from the camera plane can be transformed into an environment plane. Homography module **214** thus outputs output **224**, which connects the key points in the field plane.

For example, in a standard soccer field,  $p_{13}$  may be 50 yards away from  $p_{19}$ , and  $p_{13}$  may be 70 yards away from  $p_{18}$ . Given a plurality of such measurements, homography module **214** may map the pixel coordinates listed in table 1 to physical coordinates in a soccer field. For example, pixel coordinates (500, 500) may represent physical coordinates (50 yds, 35 yds). Because the camera angle for each image may be different, one cannot simply make a rigid one-to-one mapping of pixel coordinates and physical coordinates. Any warping, tilting, zooming, etc., captured in input image **352** needs to be properly mapped from the camera plane to a physical/field plane.

In some aspects, a different homography matrix is determined by homography module **214** for each different camera in the environment. For example, in a broadcast of a soccer match, there may be multiple different cameras situated in different locations of the soccer arena. Pose detector **204** may be used in a warm-up period to learn where landmarks are located in a given soccer field. Likewise, homography module **214** may determine a homography matrix based on the images provided by the specific camera. Subsequent to identifying the landmarks and determining the homography matrix, neural network **218** is configured to detect objects and have homography module **214** convert the pixel coordinates of the detected objects to physical coordinates using the homography matrix. For a different camera at a different position, this homography matrix will be different because the camera perspective is not identical.

In some aspects, after identifying the environment dimensions, via homography, object detector **216** is used to identify objects in the environment. Object detector **216** comprises neural network **218** and training dataset **220** in memory. Neural network **218** may be an image classifier that identifies an object in an image and outputs a label. Neural



network **218** may also be an image classifier that identifies an object in an image and generates a boundary around the object. In some aspects, generating the boundary around the object further comprises determining pixel coordinates of the object in the image. For example, the object may be a player that is identified by neural network **218**. Neural network **218** may generate the coordinates (500, 300). These coordinates may represent the center point of the boundary box generated around the identified player (i.e., the center point is a pixel that is 500 pixels away from the left-most pixel and 300 pixels above the bottom-most pixel). Subsequent to detecting objects in the image (e.g., players), homography module **214** may determine a position of the object in the environment plane (convert from the field plane). For example, homography module **214** may apply the homography matrix used to convert pixel coordinates to physical coordinates—taking input (500, 300) and outputting (50 yds, 45 yds). Homography module **214** then outputs output image **126**, which identifies object positions in the physical environment plane.

FIG. 4 illustrates a flow diagram of method **400** for determining environment dimensions based on environment pose. At **402**, object localizer **202** trains, with a dataset (e.g., training dataset **208**) comprising a plurality of images featuring an environment and labelled landmarks in the environment (e.g., landmarks **301**), a neural network (e.g., neural network **206**) to identify a pose of an environment in an arbitrary image, wherein the pose comprises connected labelled landmarks of the environment. In some aspects, the neural network comprises a convolutional backbone with feature extraction layers and a segmentation head.

At **404**, object localizer **202** receives (e.g., via user interface **212**) an input image depicting the environment. At **406**, object localizer **202** generates (e.g., via image augments **210**) an input tensor based on the received input image. At **408**, object localizer **202** inputs the input tensor into the neural network, wherein the neural network is configured to generate an output tensor comprising a position of each identified landmark, a confidence level associated with each position, and a pose confidence score. In some aspects, the confidence level is a probability of a predicted position of a landmark being correct.

In some other aspects, the confidence level is a probability of whether a respective landmark is shown in a frame of the input image. In some aspects, the neural network is configured to determine a loss that is a linear combination of a mean absolute error for each identified landmark position and a cross entropy for point presence on the frame. This loss may be optimized using stochastic gradient descent without the long range offsets found in conventional pose detection neural networks.

In some aspects, the pose confidence score is a mean value of all position confidence levels.

At **410**, object localizer **202** calculates (e.g., via homography module **214**) a homography matrix between each position in the output tensor along a camera plane and a corresponding position in an environment plane, based on a pre-built model of the environment. In some aspects, the pre-built model of the environment is indicative of distances between each landmark in the environment. At **412**, object localizer **202** outputs an image that visually connects each landmark along the environment plane based on the homography matrix.

FIG. 5 is a block diagram illustrating a computer system **20** on which aspects of systems and methods for determining environment dimensions based on environment pose may be implemented in accordance with an exemplary aspect. The

computer system **20** can be in the form of multiple computing devices, or in the form of a single computing device, for example, a desktop computer, a notebook computer, a laptop computer, a mobile computing device, a smart phone, a tablet computer, a server, a mainframe, an embedded device, and other forms of computing devices.

As shown, the computer system **20** includes a central processing unit (CPU) **21**, a system memory **22**, and a system bus **23** connecting the various system components, including the memory associated with the central processing unit **21**. The system bus **23** may comprise a bus memory or bus memory controller, a peripheral bus, and a local bus that is able to interact with any other bus architecture. Examples of the buses may include PCI, ISA, PCI-Express, HyperTransport™, InfiniBand™, Serial ATA, I<sup>2</sup>C, and other suitable interconnects. The central processing unit **21** (also referred to as a processor) can include a single or multiple sets of processors having single or multiple cores. The processor **21** may execute one or more computer-executable code implementing the techniques of the present disclosure. For example, any of commands/steps discussed in FIGS. 2-4 may be performed by processor **21**. The system memory **22** may be any memory for storing data used herein and/or computer programs that are executable by the processor **21**. The system memory **22** may include volatile memory such as a random access memory (RAM) **25** and non-volatile memory such as a read only memory (ROM) **24**, flash memory, etc., or any combination thereof. The basic input/output system (BIOS) **26** may store the basic procedures for transfer of information between elements of the computer system **20**, such as those at the time of loading the operating system with the use of the ROM **24**.

The computer system **20** may include one or more storage devices such as one or more removable storage devices **27**, one or more non-removable storage devices **28**, or a combination thereof. The one or more removable storage devices **27** and non-removable storage devices **28** are connected to the system bus **23** via a storage interface **32**. In an aspect, the storage devices and the corresponding computer-readable storage media are power-independent modules for the storage of computer instructions, data structures, program modules, and other data of the computer system **20**. The system memory **22**, removable storage devices **27**, and non-removable storage devices **28** may use a variety of computer-readable storage media. Examples of computer-readable storage media include machine memory such as cache, SRAM, DRAM, zero capacitor RAM, twin transistor RAM, eDRAM, EDO RAM, DDR RAM, EEPROM, NRAM, RRAM, SONOS, PRAM; flash memory or other memory technology such as in solid state drives (SSDs) or flash drives; magnetic cassettes, magnetic tape, and magnetic disk storage such as in hard disk drives or floppy disks; optical storage such as in compact disks (CD-ROM) or digital versatile disks (DVDs); and any other medium which may be used to store the desired data and which can be accessed by the computer system **20**.

The system memory **22**, removable storage devices **27**, and non-removable storage devices **28** of the computer system **20** may be used to store an operating system **35**, additional program applications **37**, other program modules **38**, and program data **39**. The computer system **20** may include a peripheral interface **46** for communicating data from input devices **40**, such as a keyboard, mouse, stylus, game controller, voice input device, touch input device, or other peripheral devices, such as a printer or scanner via one or more I/O ports, such as a serial port, a parallel port, a universal serial bus (USB), or other peripheral interface. A



## 11

display device 47 such as one or more monitors, projectors, or integrated display, may also be connected to the system bus 23 across an output interface 48, such as a video adapter. In addition to the display devices 47, the computer system 20 may be equipped with other peripheral output devices (not shown), such as loudspeakers and other audiovisual devices.

The computer system 20 may operate in a network environment, using a network connection to one or more remote computers 49. The remote computer (or computers) 49 may be local computer workstations or servers comprising most or all of the aforementioned elements in describing the nature of a computer system 20. Other devices may also be present in the computer network, such as, but not limited to, routers, network stations, peer devices or other network nodes. The computer system 20 may include one or more network interfaces 51 or network adapters for communicating with the remote computers 49 via one or more networks such as a local-area computer network (LAN) 50, a wide-area computer network (WAN), an intranet, and the Internet. Examples of the network interface 51 may include an Ethernet interface, a Frame Relay interface, SONET interface, and wireless interfaces.

Aspects of the present disclosure may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present disclosure.

The computer readable storage medium can be a tangible device that can retain and store program code in the form of instructions or data structures that can be accessed by a processor of a computing device, such as the computing system 20. The computer readable storage medium may be an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination thereof. By way of example, such computer-readable storage medium can comprise a random access memory (RAM), a read-only memory (ROM), EEPROM, a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), flash memory, a hard disk, a portable computer diskette, a memory stick, a floppy disk, or even a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon. As used herein, a computer readable storage medium is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or transmission media, or electrical signals transmitted through a wire.

Computer readable program instructions described herein can be downloaded to respective computing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network interface in each computing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing device.

Computer readable program instructions for carrying out operations of the present disclosure may be assembly

## 12

instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language, and conventional procedural programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a LAN or WAN, or the connection may be made to an external computer (for example, through the Internet). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present disclosure.

In various aspects, the systems and methods described in the present disclosure can be addressed in terms of modules. The term "module" as used herein refers to a real-world device, component, or arrangement of components implemented using hardware, such as by an application specific integrated circuit (ASIC) or FPGA, for example, or as a combination of hardware and software, such as by a microprocessor system and a set of instructions to implement the module's functionality, which (while being executed) transform the microprocessor system into a special-purpose device. A module may also be implemented as a combination of the two, with certain functions facilitated by hardware alone, and other functions facilitated by a combination of hardware and software. In certain implementations, at least a portion, and in some cases, all, of a module may be executed on the processor of a computer system. Accordingly, each module may be realized in a variety of suitable configurations, and should not be limited to any particular implementation exemplified herein.

In the interest of clarity, not all of the routine features of the aspects are disclosed herein. It would be appreciated that in the development of any actual implementation of the present disclosure, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, and these specific goals will vary for different implementations and different developers. It is understood that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art, having the benefit of this disclosure.

Furthermore, it is to be understood that the phraseology or terminology used herein is for the purpose of description and not of restriction, such that the terminology or phraseology of the present specification is to be interpreted by the skilled in the art in light of the teachings and guidance presented herein, in combination with the knowledge of those skilled in the relevant art(s). Moreover, it is not intended for any term in the specification or claims to be ascribed an uncommon or special meaning unless explicitly set forth as such.

The various aspects disclosed herein encompass present and future known equivalents to the known modules referred to herein by way of illustration. Moreover, while aspects and applications have been shown and described, it would be apparent to those skilled in the art having the benefit of this



## 13

disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts disclosed herein.

The invention claimed is:

1. A method for determining environment dimensions 5  
based on environment pose, the method comprising:  
training, with a dataset comprising a plurality of images  
featuring an environment and labelled landmarks in the  
environment, a neural network to identify a pose of an  
environment in an arbitrary image, wherein the pose 10  
comprises connected labelled landmarks of the envi-  
ronment;  
receiving an input image depicting the environment;  
generating an input tensor based on the received input  
image; 15  
inputting the input tensor into the neural network, wherein  
the neural network is configured to generate an output  
tensor comprising a position of each identified land-  
mark, a confidence level associated with each position,  
and a pose confidence score; 20  
calculating a homography matrix between each position  
in the output tensor along a camera plane and a corre-  
sponding position in an environment plane, based on a  
pre-built model of the environment; and  
outputting an image that visually connects each landmark 25  
along the environment plane based on the homography  
matrix.
2. The method of claim 1, wherein a camera perspective  
of the input image does not match any of the camera  
perspectives of the plurality of images in the dataset.
3. The method of claim 1, wherein the neural network  
comprises:  
a convolutional backbone with feature extraction layers,  
and  
a segmentation head. 30
4. The method of claim 1, wherein the pre-built model of  
the environment is indicative of distances between each  
landmark in the environment.
5. The method of claim 1, wherein the neural network is  
further configured to determine a heat map for each position 40  
of each identified landmark, wherein the heat map represents  
an area in which the identified landmark may be in the input  
image.
6. The method of claim 5, wherein the neural network  
optimizes a loss using stochastic gradient descent. 45
7. The method of claim 1, wherein the input image is a  
video frame of a livestream, and wherein the neural network  
determines environment dimensions in real-time.
8. The method of claim 1, wherein the environment is a  
sports field and the labelled landmarks are locations on the 50  
sports field.
9. A system for determining environment dimensions  
based on environment pose, the system comprising:  
a hardware processor configured to:  
train, with a dataset comprising a plurality of images 55  
featuring an environment and labelled landmarks in  
the environment, a neural network to identify a pose  
of an environment in an arbitrary image, wherein the  
pose comprises connected labelled landmarks of the  
environment; 60  
receive an input image depicting the environment;  
generate an input tensor based on the received input  
image;

## 14

- input the input tensor into the neural network, wherein  
the neural network is configured to generate an  
output tensor comprising a position of each identi-  
fied landmark, a confidence level associated with  
each position, and a pose confidence score;  
calculate a homography matrix between each position  
in the output tensor along a camera plane and a  
corresponding position in an environment plane,  
based on a pre-built model of the environment; and  
output an image that visually connects each landmark  
along the environment plane based on the homog-  
raphy matrix.
10. The system of claim 9, wherein a camera perspective  
of the input image does not match any of the camera  
perspectives of the plurality of images in the dataset.
  11. The system of claim 9, wherein the neural network  
comprises:  
a convolutional backbone with feature extraction layers,  
and  
a segmentation head.
  12. The system of claim 9, wherein the pre-built model of  
the environment is indicative of distances between each  
landmark in the environment.
  13. The system of claim 9, wherein the neural network is  
further configured to determine a heat map for each position  
of each identified landmark, wherein the heat map represents  
an area in which the identified landmark may be in the input  
image.
  14. The system of claim 13, wherein the neural network  
optimizes a loss using stochastic gradient descent.
  15. The system of claim 9, wherein the input image is a  
video frame of a livestream, and wherein the neural network  
determines environment dimensions in real-time.
  16. The system of claim 9, wherein the environment is a  
sports field and the labelled landmarks are locations on the  
sports field.
  17. A non-transitory computer readable medium storing  
thereon computer executable instructions for determining  
environment dimensions based on environment pose,  
including instructions for:  
training, with a dataset comprising a plurality of images  
featuring an environment and labelled landmarks in the  
environment, a neural network to identify a pose of an  
environment in an arbitrary image, wherein the pose  
comprises connected labelled landmarks of the envi-  
ronment;  
receiving an input image depicting the environment;  
generating an input tensor based on the received input  
image;  
inputting the input tensor into the neural network, wherein  
the neural network is configured to generate an output  
tensor comprising a position of each identified land-  
mark, a confidence level associated with each position,  
and a pose confidence score;  
calculating a homography matrix between each position  
in the output tensor along a camera plane and a corre-  
sponding position in an environment plane, based on a  
pre-built model of the environment; and  
outputting an image that visually connects each landmark  
along the environment plane based on the homography  
matrix.

\* \* \* \* \*